UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/735,715	12/12/2000	Jacob Dreyband	033144-004	5590

23562 7590 04/12/2006

BAKER & MCKENZIE LLP
PATENT DEPARTMENT
2001 ROSS AVENUE
SUITE 2300
DALLAS, TX 75201

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 04/12/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/735,715	DREYBAND ET AL.	
	Examiner	Art Unit	
	Tuan A. Vu	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 February 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-7,9-21 and 43-71* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-7,9-21 and 43-71* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 2/17/2006.

As indicated in Applicant's response, claims 1, 43 have been amended, and claim 8 canceled. Claims 1-7, 9-21, 43-71 are pending in the office action.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Claims 1-21, 43-49, and 56-67 are rejected under 35 U.S.C. 112, first paragraph, as based on a disclosure which is not enabling. The disclosing of steps for packing of data having particular length, and type being associated with a value or an identifier in a tagging scheme -- wherein a sequence is allocated by a memory size value encapsulated under a header value -- as exemplified via Fig. 3-5 as well as the process of unpacking of Fig. 7-9 are critical or essential to the practice of the invention, but not included in the claim(s) is not enabled by the disclosure. See *In re Mayhew*, 527 F.2d 1229, 188 USPQ 356 (CCPA 1976).

Specifically, both claims 1 and 43 recite " ... packing the tagged data as a binary representation of the tagged data object that is transferable among, and processible by, without any intermediate data format conversions, any computer processing unit processing data..." (*). As recited, this limitation is interpreted according to the context wherein the packed object has been the result of the earlier steps recited as 'creating a tagged data object...; encapsulating a data element ... includes a data element and a corresponding binary tag id'. The mere facts of creating tagged data object in a container to provide computer-based 'universal manipulation and

aggregation of structure and unstructured data' by further encapsulating in the tagged data object a data element with a corresponding binary tag id would not enable one skill in the art to construe how this tagged data by being 'platform, hardware, or language independent', as claimed, to be sufficiently universal for the tagged data, when transferred as in a binary representation would become presentable to and processible by any computer unit without any intermediate format conversions. It is noteworthy to indicate that the phrase recited as 'for use by an application that is prescribed ... any computer environment' will be treated as an intended use and would bear no weight on the patentability analysis of the method steps leading to so-called tagged data being 'processible without any intermediate format conversions'. For one skill in the art, the mere fact of encapsulating data under an identifier would at least entail a form of re-extracting such tagged data from the encapsulated form by a computer receiving this data, whether or not the data element inside the tagged object (and its the tag id) were in a binary form. The disclosure depicted via Figures 7-9 has it made clear that some unpacking process is to be effectuated, otherwise, so it is implied, the tagged data would remain unusable for being still encapsulated; hence a form of readjustment/extraction of data operated upon the tagged object being received as encapsulated form has occurred, i.e. without which it cannot be presentable for use. There appears that the claimed 'tagged data object processible by, without any intermediate format conversions, any computer ...' does not teach or reasonably convey the underlying mechanics enable one skill in the art to make use of or construe such tagged data in a way that it is so processible as recited, given the encapsulation step and the broad interpretation coming from the recital that a container is platform, hardware and language independent; when there is no specificity in the claim language about this universal container to shed a reasonable

insight as to how this claimed multi-independency or such encapsulating can support the waiving/bypassing of intermediate format conversion step as mentioned above. In the BACKGROUND of the invention (Specifications - pg. 1-2), it is conveyed that HTML, ASCII, JDOM form of data has its disadvantage because it is confined to a particular platform that is prescribed to understand just these form; and there is a need to provide a document model that may be used universally among languages. The claim only recites a global assertion that the container in the tagged data is universal and by further providing encapsulating the data element inside the tagged object with a binary id, the claim appears to leap right to the point where when the tagged object is transferred to any computer, where it can be processed without any intermediate reformatting or reconverting. It was a well known concept that data are tagged from transmission and do include header with id and streamed in a network/protocol based binary form to be un-marshaled or unpacked at specific levels of the network layers to become available in a form that the higher NW layer can interpret and make use of. The likes of which layers happen to be, *inter alia*, an Application layer wherein data are presented in a format understandable by the Application layer albeit the fact that when the data stream gets unpacked at the lower level, they are binary per nature of the Network protocol. The claim clearly omits essential steps that would enable how the encapsulated tag id of and the data element included in universal container would help the use (by any processing unit) of such tagged object without a form of unpacking or extracting of data within the received tagged object given the commonly accepted knowledge that there always has been a transmitted network data being processed upon receipt, such processing being integral to each layer of network ISO hierarchy. The claim describes what appears to be a truncated scenario of packing data while the disclosure depicts

another scenario enabling a more elaborate embodiment thereof; thus in view of both scenarios, the claim limitation as mentioned above (see *) is not enabled by the specifications for lacking what appears to be essential steps that would reasonably convey how data can be processible without reformat or reconversion.. In short, claims 1 and 43 are hence rejected for not being enabled by the specifications in light of the above. This limitation would be treated as though some tagged data will be processible by any computer, while the intermediate conversion would bear the limited connotation to the extent as far as parts the disclosure have allowed (see Fig. 7-9), that is, the conversion would be a form of unpacking.

Dependent claims 2-21, 44-49, and 56-67 are also rejected for not remedying to the deficiencies of the base claims.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-7, 9-21, and 43-71 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bowman-Amuah, USPN: 6,477,580 (hereinafter Bowman), in view of Francis et al., USPN: 6,665,861 (hereinafter Francis), and further in view of White et al., 6,438,559 (hereinafter White).

As per claim 1, Bowman discloses a method for presenting data within a computing environment including an application program interface (e.g. col. 103, line 63 to col. 105, line 6;

col. 36-38 – Note: window based application implicitly discloses APIs), said method comprising the steps of:

creating a tagged data object for storing data having a container(e.g. *software package* – col. 105, line 42-66; *bundle, message* – Fig. 185-187; Fig. 98 – Notes: browser interpreting pages is equivalent to tagged data being stored in message or packages streamed between browser applications or interconnected framework machines; a message or package being analogized to a container);

encapsulating a data element into a tagged data object to provide a tagged data including a data element (e.g. *encapsulation* – col. 299, line 1 to col. 300, line 14; Fig. 98; Fig. 184-185);

packing the tagged data by converting the tagged data as a binary representation of tagged data object (e.g. Fig. 20-22; *stream out business object* – col. 281, line 17 to col. 282, line 29; Fig. 165; data *stream* -Figs. 184-191; *reuse, binaries, beans* – col. 181, lines 39-57- Note: a stream being passed over the internet reads on binary representation of being packed tagged data being streamed and eventually processed by recipient machine application engines) that is transferable among, presentable to, and processible by any computer processing unit, such object being processible without other intermediate data conversion (Note: data encapsulated in messages and processed via unpacking by lower network layers and channeled up to and for use in the application layer reads on 'without any other intermediate format conversion', i.e. wired/binary stream from lower socket layer via inherent Network layers processing/unpacking for use at the higher Application layer destination according to Bowman's framework – see Fig. 20-25);

such processing unit being prescribed by a data conversion and a formatting specification and operable in a computer environment (Note: stream of data being processed within the network of layers so that the format being operable at the API level such as that of Bowman's paradigm – see Fig. 42 – read on tagged object presentable and processible by any processing unit prescribed for the GUI interface formatting and application -- e.g. browser, case tool, COM middleware -- conversion thereof);

But Bowman does not explicitly specify a tagged data object as a message/bundle container form such that this container is universal tagged data object being platform independent, hardware independent, and language independent. But in view of Bowman's disclosing of COM format for effecting RPC, messaging utilities and directory services having platform independent standard for transmitting data (e.g. Fig. 20-22; col. 73, lines 10-41; col. 63, line 62 to col. 63, line 21 – Note: COM format data are platform and language neutral by nature of common platform object broker services), in combination of language neutral for Java byte codes (*virtual machines 2706* – Fig. 27), and binary representation across hardware independent internet protocol, the above limitation is at least strongly suggested. The packaging of data using Java platform neutral format was a well-known concept in the art of software transmission at the time the invention was made. Francis, in a method to transmit package of Java binary representation and metadata similar to Bowman streaming of data (Bowman: Fig. 108-109) across computers, also discloses packaging binary representation of Java beans with supporting utilities/metadata contained in a markup file as markup or tagged form like XML (Fig. 6-8). In case the platform, hardware and language neutral package stream by Bowman are not universal tagged data for browser use, it would have been obvious for one of ordinary skill in the art at the

time the invention was made to encapsulate such data in container being in XML form as taught Francis because this will alleviate resources of the receiving computer in making use of readily formatted data without additional compilation.

Further, Bowman discloses that the above universal tagged data container being encapsulated for universal manipulation and aggregation by computer processing units according to the scenario associated with network stream of binary packets across platform, which discloses encapsulated data for universal access, universal in the context of many machines that can establish reception of such stream. As for the access to manipulation and aggregation of tagged data, Bowman discloses browser manipulation of tagged data and markup language data manipulation and aggregation using browser application in conjunction with stream, message passing and ORB remote calls using platform independent-based services as mentioned above (e.g. Fig. 13-18; Fig. 98) and processing of wired/binary stream of data (Fig. 22-25). Thus Bowman has disclosed access of tagged (or markup) data for manipulation or aggregation (Note: data provided through COM or ORB services and a compilation of HTML formatted data in pages composed of subdivided markup sections implicitly disclose access for manipulation or aggregation of data; further the manipulation of wired data arriving at the socket layer being parsed and processed so that they become ready to use at higher layer reads on manipulation and aggregation by computer unit as set forth above).

But Bowman does not explicitly specify that the encapsulated tagged data includes data element and a corresponding binary tag id. Network data being streamed encompasses header to represent some content of a packet/message container, and Bowman HTML-based document passing entails markup with header and Id representing a tagged content. Hence, Bowman or

Francis, teaches tagged data for interpretation by browsers, hence implicitly discloses a variable name bracketed within the begin tag and end tag; and teaches attributes descriptors in the meta-data section comprising a header section of the object-based stream, which suggests encapsulating identification information of the data element of the stream, according to a well-known concept of including some identifier in a binary form to associate the data bundle or packet sent over a network communication link with its content at the time the invention was made. White, in a method to serialize objects for distribution over a communication network environment using descriptors in serialization of class objects analogous to the object streaming and meta-data by Bowman or Francis, discloses the tagging of object content being serialized with an identifier or value for packing and de-serializing (e.g. *ACI* - col. 4, lines 16-58; col. 10, line 36 to col. 11, line 9). It would have been obvious for one of ordinary skill in the art at the time the invention was made to use the tagging technique associating an identifier with the tagged content as taught by White and apply it to the stream metadata by Bowman, in case Bowman's metadata or tagged stream does not include such tag identification already, because this tag ID would facilitate the differentiation between data being packed and enable data handling/re-processing as well as unpacking or modification of elements packed in the message or bundle.

As per claim 2, Bowman discloses the packing of tagged data being a simple object and a complex object, and list object (e.g. col. 124, line 14 to col. 127, line 39 – Note: the use of Java or C++ based components implicitly discloses basic class, compound classes, or structure/enumeration of basic classes and compound classes objects).

As per claim 3, Bowman discloses packing a simple object by retrieving data attributes for length of an object source identifier, object size, type, value; allocating of packed memory location for object identifier length (e.g. col. 235, line 47 to col. 237, line 32); copying the object size, type, and value into the packed memory location (Note: this is inherent to the above cited portions); retrieving and copying head value and exit value into the packed memory location (e.g. *START INDEX, WS-INDEX, STREAM-END* - col. 237, line 35 to col. 238, line 66).

As per claims 4 and 5, Bowman does not explicitly specify the steps of retrieving, writing, and allocating/writing for the complex object as has been disclosed for the simple object from claim 3, but in view of the packaging of data in the retrieval of business-related complex object (e.g. col. 204, line 40 to col. 207, line 59), the limitations as recited are herein implicitly in view of the inherent presence of simple object within complex object or list objects.

As per claim 6 and 7, Bowman does not specify packing list object with retrieving of object source identifier, allocating memory in a packed memory location to accommodate the list object source identifier length; retrieving and copying list head value and list exit value into the packed memory location; but in view of the rationale used in addressing claims 4 and 5, these limitations are also implicitly disclosed because of the inherent presence of simple objects and complex objects in structure or enumeration, i.e. list, object so well-known in object-oriented language.

Further, Bowman does not explicitly disclose retrieving list array object and copying it to the packed memory location. But, in view of the inherent array structure in structure or enumeration of simple and complex objects in C++ or Java, this limitation is also implicitly disclosed as per the same rationale used for claims 4 and 5.

As per claim 9, see Bowman (e.g. Fig. 20-22; *stream out business object* – col. 281, line 17 to col. 282, line 29; Fig. 165; data *stream* -Figs. 184-191).

As per claim 10, refer to claim 2.

As per claim 11, Bowman discloses data wrapping (e.g. *Wrapper component* - Fig. 81).

As per claim 12, Bowman (col. 131-132; col. 174, line 33 to col. 175, line 24; Fig. 50-51), discloses modeling using COM and Case Tools but Bowman does not specify including of named tree with a field name connected with a value. But in view of the teaching of language-independent modeling along with metadata or language neutral format as addressed in claim 1, (Francis' teaching modeling and tagged web format data is suggesting of tree structure implementation of data to be transmitted as metadata or specification data), this limitation would have obvious for the same rationale as used in claim 1 and also the association of tree with field name as metadata would enhance the utilization and re-processing of data tagged and stored in the package.

As per claim 13, Bowman discloses Java and C++ constructs which inherently include list or enumeration of objects of simple and complex type (see claim 2).

As per claim 14, this claim includes the encapsulation of data type, tag id, and writing thereof to the tagged data object and these limitations have been addressed in claim 3 and 4.

As per claim 15, Bowman discloses the use of Java objects, hence has implicitly disclosed one of the following data type: integer, float, byte, char string, a java object, a null data, a primitive type, a compound type, and a list type.

As per claim 16, Bowman (in combination with White/Francis) discloses tag identifier with of type integer (see White from claim 1).

As per claim 17, Bowman with White's teachings discloses serializing of tagged data and compacting it in a stream for transmission, hence has implicitly disclosed a tagging process following a linear sequence, i.e. sequential tagging with determining a sequence.

As per claim 18, Bowman with White's teachings discloses including a data, a position and a tag element (refer to claim 3; Fig. 109 – Note: Index position use in writing data by Bowman discloses including an position and packet layout inherently encompasses boundaries position of data compacted in packet).

As per claim 19, Bowman does not explicitly specify converting of first type of tagged data to second type of data for a change in properties; but the concept for converting the order of data type (e.g. network-bound integer converted into local host-based integer and vice-versa, as per Java/C++ *ntohs* or *htons* functions) for allowing data type to be communicated through the internet medium was a well-known concept at the time the invention was made. Hence, Bowman's disclosed communication of Java or C++ objects implicitly discloses such conversion to provide for a communication properties adjustment or change as claimed.

As per claims 20 and 21, by virtue of the rejections of claim 2 and claim 19 above, the limitations of these claims are implicitly disclosed.

As per claim 43, Bowman discloses a method for presenting data within a computing environment including an application program interface prescribed for data conversion and wire formatting specification (e.g. col. 103, line 63 to col. 105, line 6; col. 36-38 – Note: window based application implicitly discloses APIs), said method comprising the steps of:

creating a tagged data object comprising a tagged data being a container (e.g. *software package* – col. 105, line 42-66; *bundle, message* – Fig. 185-187; Fig. 98);

encapsulating a data element into a tagged data object to provide a tagged data including a data element (e.g. e.g. *encapsulation* – col. 299, line 1 to col. 300, line 14; Fig. 98; Fig. 184-185);

providing the tagged data by converting the tagged data as a binary representation of tagged data object (e.g. Fig. 20-22; *stream out business object* – col. 281, line 17 to col. 282, line 29; Fig. 165; data *stream* -Figs. 184-191; *reuse, binaries, beans* – col. 181, lines 39-57);

transmitting the tagged data transmission (e.g. Fig. 105-107);

unpacking the tagged data from a binary representation; creating tagged data object for storing the tagged data; and extracting a data element for the tagged data (e.g. Fig. 185, 187; *unpackaged* - col. 300, line 39 to col. 301, line 29; col. 181, lines 39-57).

Bowman further discloses tagged data object to provide universal access to manipulation and aggregation by computer processing units of a structured data and unstructured data (e.g. Fig. 13-18; Fig. 98; Fig. 22-25 --Note: refer to rationale for this limitation as it has been set forth in claim 1 and 8 above).

Bowman discloses that the packed tagged data object such that this tagged object is transferable to, presentable to, and processible by any processing unit without other intermediate data conversion (Note: data encapsulated in messages and processed via unpacking by lower network layers and channeled up to and for use in the application layer reads on 'without any other intermediate format conversion', i.e. wired/binary stream from lower socket layer via inherent Network layers processing/unpacking for use at the higher Application layer destination according to Bowman's framework – see Fig. 20-25);

such processing unit being prescribed by a data conversion and a formatting specification and operable in a computer environment (Note: stream of data being processed within the network of layers so that the format being operable at the API level such as that of Bowman's paradigm – see Fig. 42 – read on tagged object presentable and processible by any processing unit prescribed for the GUI interface formatting and application -- e.g. browser, case tool, COM middleware -- conversion thereof).

But Bowman does not explicitly specify a tagged data object as a message/bundle container form such that this container is universal tagged data object being platform independent, hardware independent, and language independent. But this limitation has been addressed in claim 1; nor does Bowman explicitly specify that the encapsulated tagged data object or container includes a data element and a corresponding tag id; but this also has been addressed in claim 1 above using Francis/White.

As per claims 44-49, these claims correspond to claims 2-7 respectively; hence are rejected likewise, respectively.

As per claim 50, this corresponds to claim 2, and is rejected using the rationale of claim 2. Further, Bowman discloses a method for presenting data within a computing environment including an application program interface comprising the steps of unpacking tagged data from a binary representation; creating tagged data object for storing the tagged data; and extracting a data element for the tagged data (e.g. Fig. 185, 187; *unpackaged* - col. 300, line 39 to col. 301, line 29-- Note: in view of the teachings on packing data into package or stream to be sent in packet over the internet by Bowman as mentioned in claim 1, the steps of unpacking, creating a

storage for the unpacked data received over the internet, and the extracting of object being tagged are implicitly disclosed).

As per claim 51, Bowman does not specify the steps of retrieving the simple head value and simple exit value; allocating memory in an unpacked memory and copying of simple object size, type and value into said unpacked memory. Subjecting packets received from the internet/network into a host or routing, or a gateway machines to unpacking and buffer storage was a well-known concept in the art at the time the invention was made. In view of the teachings for unpackaging of data by Bowman above and the well-known unpacking of data, it would have been obvious for one skill in the art at the time the invention was made to provide the unpacking of the tagged data as taught by Bowman/Francis/White using the well-known technique of unpacking/storage above because this would enable correct extraction of data based on boundaries locations and allocation of correct memory resources.

As per claims 52 and 53, the limitations as to unpack a complex object would also have been obvious by virtue of the inherency of simple object in a complex objects as mentioned in claims 4-5 and the rejection used in claim 51 above.

As per claims 54 and 55, the rationale used for claims 6-7 and 52-53 are herein applied.

As per claims 56-59, refer to rejections of claims 10-13 respectively.

As per claim 60, this claim corresponds to claim 14, hence is rejected using the same rationale as set forth therein.

As per claims 61-67, refer to corresponding rejections of claims 15-21 respectively.

As per claim 68, Bowman does not specify extracting data with determining the type to provide the tag id; and writing the data element into the tagged data object. But in view of

White's or Francis's teaching to provide a tagging associated with an identifier in order to facilitate the reprocessing of data manipulated at the receiving end and the rationale for encapsulating in claim 14, this step would have been obvious because the implied and inherent association between packing and unpacking.

As per claims 69 and 70, see rejection of claims 15 and 16 respectively.

As per claim 71, in view of the unpacking as taught by Bowman and the rationale in claim 18 above, this limitation would also have been obvious by virtue of the adding of element in the tagged data as mentioned in the above rejection.

Response to Arguments

6. Applicant's arguments filed 2/17/06 have been fully considered but they are not persuasive. Following are the Examiner's observations in regard thereto.

As per the Rejection under 35 USC §112:

(A) Applicant has submitted that by amending that the tagged data is presentable to and processible to a processing unit (CPU) with a uncommon approach as to obviate the conversion of a application-specific data to a binary format (Appl. Rmrks, pg. 17, bottom); thus permitting an application prescribed for a formatting specification to process the data element in the tagged data because of the universal nature of the container. The statement appears to present some contradicting teaching as a whole, i.e. processible by an application specific to some formatting versus universal nature data processible by any computer; and conveys a concept not commonly admitted in all software technologies, that is, producing a binary representation for use in an application with that binary being converted necessarily from an application-specific language; when in fact text like ASCII or HTML can be processed in any browser. But in view of the focus

being imparted to only what is being claimed and the observations set forth in the current rejection under USC 112 1st paragraph, the deficiency of the above statement is not addressed and the lack of essential teachings in the claims – or deficiency thereof -- would have to be referred to the subject being discussed as set forth in said USC 112 rejection.

As per the Rejection under 35 USC §103:

(B) Applicant has submitted that in view of the change to claims 1 and 43, Bowman for disclosing browsers employing HTML, XML formats to process data received does not disclose 'binary code executable' by CPUs as recited in the amended claims; nor does Bowman disclose universal tagged object that provides universal ... by any computer units (Appl. Rmrks, pg. 19, bottom). The rejection has addressed what constitutes a binary representation, what amounts to a universal manipulation and aggregation, what can be perceived as a tagged data in a container being streamed as a wired format to be funneled from the lower network hierarchy layer to the higher application layer via manipulation and aggregation, let alone the teaching that Bowman does deliver data in executable form like beans, or binaries. Further, the term *binary code executable* is not found in the claims. Granted that a binary representation can be an executable, which is not clear from the claim, a binary being provided as executable code to an application layer for use would not be considered universal tagged data immediately presentable to or processible without being converted by intermediate format. Network data arriving at a socket level was known to have been converted, parsed or extracted to meet a form required for use in an application environment prescribed to execute such form would be known; and a package or packet containing a executable code would be subject to same. Besides, it would be very hard to demonstrate which part of data being transmitted or received is actually binary or not binary

when the claim does not provide unequivocal teaching as to how this packed/tagged data constitutes; e.g. that by *binary representation*, it does mean something very specific and distinctly formatted unlike the likes of streams (e.g. of 0's and 1's) perceived in Bowman's data contained in message or encapsulated objects of a NW bandwidth. Further, in light of the rejection set forth in the USC 112, 1st paragraph, some limitations have not been given full patentable weight. Thus, for some features, the rejection has used interpretation based in part on the specifications in regard to those; and/or broad reasonable interpretation thereof to address the amended claims including all those limitations being slighted as a result of said USC 112 rejection. The rejection has set forth each cited parts of Bowman to map each of the claimed features; or has combined teachings to address others that are considered obvious. It is Applicant's responsibility to clearly point out where exactly the teaching by the cited parts of Bowman have not appropriately anticipating the claimed features, and/or rendered them obvious in light of the secondary references like Francis or White. Applicant's maintaining that Bowman does not disclose or suggest a particular feature (*binary representation* -- Appl. Rmrks, pg. 20, bottom) when in fact such feature has been mapped by Bowman in the rejection (refer to claim 1: Fig. 22-25) would not support Applicant's rebut or would not impart an otherwise patentable value to the claimed features because that would amount to mere allegation without legitimate and corroborative support. In other words, Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

For the above reasons, the claims stand rejected as set forth in the Office Action.

Conclusion

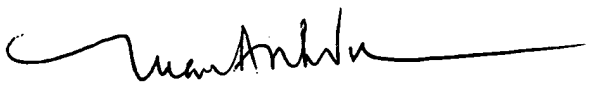
7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Tuan A Vu
Patent Examiner,
Art Unit 2193
April 10, 2006